



# Pizzamania

Link: <http://www.spoj.com/problems/OPCPIZZA/>

Solution:

C++: <http://ideone.com/q7Hpy5>

Python: <https://ideone.com/5ABY3O> (Code này dùng để tham khảo, khi submit sẽ bị TLE do dữ liệu python chưa đọc phù hợp)

Java: <https://ideone.com/Cho5xp>

**Tóm tắt đề:** Có  $N$  người muốn ăn pizza, họ chia thành từng cặp để cùng nhau mua pizza, một cặp chỉ có thể mua được pizza nếu tổng số tiền của hai người đúng bằng giá của chiếc bánh pizza. Hỏi số cặp tối đa có thể ghép thành để mua được nhiều bánh pizza nhất, mỗi người chỉ thuộc tối đa một cặp.

## Input:

Dòng đầu tiên chứa số nguyên  $T$  là số test case. Tiếp theo là các test case với cấu trúc sau:

- Dòng đầu chứa 2 số nguyên dương  $N$  là số người và  $M$  là giá của một chiếc bánh pizza.
- Dòng thứ hai chứa  $N$  số nguyên là số tiền mà người thứ  $i$  đang có.

Giới hạn:  $T \leq 100$ ,  $1 \leq N \leq 10^5$ , các giá trị còn lại nằm trong giới hạn kiểu int của C/C++ (tức là số nguyên 32 bit). Số tiền của mỗi người đều phân biệt.

## Output:

Với mỗi test case, xuất ra trên một dòng là số cặp tối đa ghép được.

2	1
4 12	1
9 -3 4 3	
5 -9	
-7 3 -2 8 7	

## Giải thích:

- Ở test 1, chỉ một cặp có thể tạo được là  $\{9, 3\}$ .
- Ở test 2, chỉ có một cặp tạo được là  $\{-7, -2\}$ .

**Hướng dẫn giải:** Ở đây đề bài yêu cầu ta chia  $N$  số nguyên thành nhiều cặp nhất và mỗi cặp có tổng bằng  $M$ . Vì mọi giá trị trong mảng đều phân biệt, nên 2 cặp  $\{a, b\}$  và  $\{c, d\}$  với  $a+b = c+d = M$  thì ta chắc chắn thấy rằng 4 số này đều sẽ khác nhau. Như vậy có thể sử dụng binary search. Với mỗi giá trị  $x$ , ta sẽ thử tìm  $y$  sao cho  $x+y = M$ , vì vai trò của  $x$  và  $y$  như nhau, nên ta giả sử  $x < y$ , như vậy, bài toán sẽ trở thành với mỗi phần tử  $x$  ta chỉ cần kiểm tra xem phần tử  $M-y$  có nằm trong mảng hay không. Tức duyệt lần lượt qua mỗi phần tử, với mỗi  $a[i]$ , thì xem  $M-a[i]$  có xuất hiện trong mảng con từ  $i+1$  đến  $N-1$  hay không (index tính từ 0).

Big-O Coding